

Reliable Final Computational Results from Faulty Quantum Computation

Gerald Gilbert, Michael Hamrick, Yaakov S. Weinstein*

Quantum Information Science Group

MITRE

260 Industrial Way West, Eatontown, NJ 07724 USA

In this paper we extend both standard fault tolerance theory and Kitaev's model for quantum computation, combining them so as to yield quantitative results that reveal the interplay between the two. Our analysis establishes a methodology that allows us to quantitatively determine design parameters for a quantum computer, the values of which ensure that an overall computation of interest yields a correct *final result* with some prescribed probability of success, as opposed to merely ensuring that the desired *final quantum state* is obtained. As a specific example of the practical application of our approach, we explicitly calculate the number of levels of error correction concatenation needed to achieve a correct final result for the overall computation with some prescribed success probability. Since our methodology allows one to determine parameters required in order to achieve the correct final result for the overall quantum computation, as opposed to merely ensuring that the desired final quantum state is produced, our method enables the determination of *complete* quantum computational resource requirements associated to the actual solution of practical problems.

PACS numbers: 03.67.Lx, 03.67.Pp

I. INTRODUCTION

The purpose of any computation, whether implemented by a quantum computer or a classical computer, is to compute the value of a function for specified values of the variables on which the function depends. In the case of a quantum computer, the final result is obtained from the outcome of a final measurement performed on the final quantum state produced by the quantum computation. This is illustrated schematically in Figure 1, which also serves to fix the terminology we shall use in this paper to describe the quantum computation and the final measurement. In particular, in this paper, the term “quantum computation” very specifically refers solely to the dynamical evolution of the qubits from some initial quantum state to some final quantum state. Our use of this term does *not* include the subsequent, final measurement from which the final result of the overall computation is obtained. We thus distinguish between the *final*

quantum state of the *quantum computation* and the *final result* of the *overall computation*, of which the quantum computation is merely a part.

Since the final measurement produces, in general, a probabilistically distributed set of outcomes, the question arises as to whether or not reliable *final* results for the overall computation can be obtained. An affirmative answer to this question is necessary in order that algorithms such as Shor's algorithm [1] can be successfully applied to practical problems. A partial answer to this question follows from the quantum computational model formalized by Kitaev ([2], §4.1). Kitaev's model identifies a bound, p , on the probability that the final result of the overall computation is incorrect due to the indeterminism of the final measurement. It follows that reliable overall computation can be achieved for sufficiently small p . However, Kitaev's model assumes that the quantum computation that produces the final quantum state (on which the final measurement is performed in order to give the final result of the overall computation) is perfectly implemented, with no errors. In other words, those problems that are within the purview of fault tolerance analysis, namely, the occurrence of errors in the *quantum* computation, and the effectiveness of error correction techniques in reducing the effect of such errors on the final *quantum* state, are not addressed in Kitaev's model. There are thus two sources of error that may affect the final result of the overall computation: (1) errors which arise in the course of the quantum computation, and (2) errors due to the indeterminacy intrinsic to the final measurement which follows the quantum computation [3]. Fault tolerance theory [4, 5, 6] addresses the first type of error, but does not consider the second. Kitaev's model addresses the second type of error, but not the first. In the present paper, we extend both fault tolerance theory and Kitaev's model so as to take into account the combined

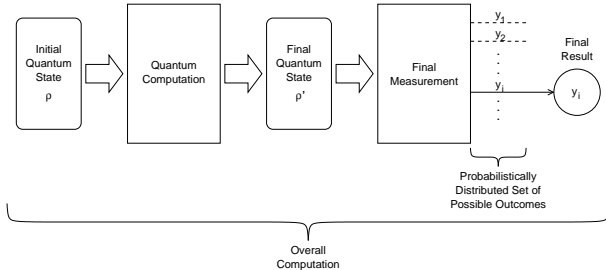


FIG. 1: Basic definitions.

*Electronic address: ggilbert, mhamrick, weinstein@mitre.org

effects of both sources of error, including the interplay between them.

In order to quantify the connection between fault tolerance theory and Kitaev's model, we utilize a measure of the difference between the desired final quantum state (that would arise in the case of perfect quantum computation) and the actual final quantum state (that is produced in an actual quantum computation realized by a practical device). We define this measure in terms of a suitable norm of the difference between the two aforementioned states. We refer to this measure as the "implementation inaccuracy." We then introduce a fundamental inequality, to which we refer as the Quantum Computer Condition (QCC), which requires that the implementation inaccuracy be less than some prescribed bound. This fundamental inequality furnishes a criterion for successful fault tolerant quantum computation. We shall see that the use of this criterion allows the conditions that ensure fault tolerance to be incorporated into Kitaev's model in a remarkably straightforward way.

Our results provide a quantitative relationship between fault tolerance theoretic constraints on the one hand, and the accuracy of the final result of the overall computation on the other. Consequently, our results can be applied to explicitly and directly determine requirements on practical fault tolerant design parameters in order to achieve a specified accuracy for the *final result* of the overall computation. This goes beyond standard fault tolerance theory, which only addresses the accuracy with which the quantum computation realizes the desired *final quantum state*, and does not directly determine the accuracy of the final result of the overall computation. As an example of the application of our result, we show how to explicitly calculate the number of levels of concatenation of error correction required to directly achieve a specified probability that the overall computation produces the correct *final result*, and not merely the desired final quantum state.

This paper is organized as follows. In Section 2 we review the Kitaev model and re-express it in a form that is convenient for our analysis. In Section 3 we introduce the implementation inaccuracy and rigorously define the fundamental inequality known as the QCC. In Section 4 we derive the constraints implied by combining the QCC with the Kitaev model. As an example of the practical utility of our general result, we explicitly calculate the number of levels of error correction concatenation needed to achieve a correct final result for the overall computation with some prescribed success probability. In Section 5 we present our conclusions.

II. THE KITAEV MODEL

We begin by reviewing the Kitaev model [2], which can be depicted by the following diagram:

$$\begin{array}{ccc}
 H_{\text{logical}} & \xrightarrow{U} & H_{\text{logical}} \\
 \uparrow \text{I}_{\text{pure}} & p & \downarrow \text{O}_{\text{pure}} \\
 X & \xrightarrow{F} & Y
 \end{array} \quad (1)$$

This diagram expresses the fact that the output of the quantum computation U is intended to be used in computing the function F , where $F : X \rightarrow Y$ is an instance of the overall computational problem. Here I_{pure} is an initialization map, which maps the input space X into pure states of the logical Hilbert space H_{logical} , and O_{pure} is the corresponding readout map, which maps the output of the operation U onto the output space Y . In general O_{pure} is a measurement given by a projection-valued measure (or more generally a POVM) $\{E_y\}_{y \in Y}$. The symbol in the center of the diagram denotes the probabilistic inaccuracy, p , associated to the output of the overall computation. If $p = 0$, the computation always produces the correct result, and the diagram commutes. Thus, the quantity $1 - p$ furnishes a lower bound on the probability of the success of the overall computation. We note that p may be non-vanishing, even if the quantum algorithm is perfectly implemented, due to the fact that the measurement of the final quantum state produced by the quantum computation is necessarily quantum probabilistically distributed. Given an input $x \in X$, the final result of the overall computation is distributed according to the probability measure on Y as follows: the probability $\text{Pr}_x(y)$ of a singleton $y \in Y$ is $\langle \text{I}_{\text{pure}}(x) | U^\dagger E_y U | \text{I}_{\text{pure}}(x) \rangle$. Then, the *near commutativity* of Diagram (1) means that for each $x \in X$, the probability measure $\text{Pr}_x(y)$ is concentrated at $y = F(x)$, so that $F(x)$ is the final result of the overall computation with high probability. Thus, Kitaev's formulation [2] requires that the Diagram (1) be nearly commutative in a probabilistic sense, that is

$$\langle \text{I}_{\text{pure}}(x) | U^\dagger E_{F(x)} U | \text{I}_{\text{pure}}(x) \rangle > 1 - p. \quad (2)$$

In other words, the measurement of the final quantum state gives the correct final result for the overall computation with probability greater than $1 - p$. If p is sufficiently small, *e.g.*, $p < 1/2$ in the case of Y binary, a majority vote algorithm will successfully identify the correct outcome $y = F(x)$.

We wish to extend the Kitaev model to apply to the scenario in which errors may occur in the implementation of the quantum computation. As a preliminary step, we re-express the Kitaev model in a more convenient form for our purposes by generalizing Diagram (1) and eq.(2) through the replacement of wavefunctions with density matrices. This allows us to discuss the effect of errors

that transform pure states into mixed states. The generalized diagram is

$$\begin{array}{ccc} \mathbf{T}(H_{\text{logical}}) & \xrightarrow{G} & \mathbf{T}(H_{\text{logical}}) \\ \text{I}_{\text{dm}} \uparrow & p & \downarrow \text{O}_{\text{dm}} \\ X & \xrightarrow{F} & Y \end{array} \quad (3)$$

where the action of the map G is given by $G : \rho \mapsto U\rho U^\dagger$, ρ is the density matrix representing the initial state of the quantum computer, the action of the map I_{dm} is given by $\text{I}_{\text{dm}} : x \mapsto |\text{I}_{\text{pure}}(x)\rangle\langle\text{I}_{\text{pure}}(x)|$, O_{dm} is the measurement corresponding to O_{pure} , except that O_{dm} acts on density matrices, and $\mathbf{T}(H)$ is the Banach space of trace class operators on a given Hilbert space, H . The probability $\text{Pr}_x(y)$ of a singleton $y \in Y$ is now $\text{tr}(\sqrt{E_y}U\text{I}_{\text{dm}}(x)U^\dagger\sqrt{E_y})$. The near commutativity of Diagram (3) means that

$$\text{tr}(\text{O}_{\text{dm}}(G(\text{I}_{\text{dm}}(x)))) \equiv \text{tr}\left(\sqrt{E_{F(x)}}U\text{I}_{\text{dm}}(x)U^\dagger\sqrt{E_{F(x)}}\right) > 1 - p \quad (4)$$

for all $x \in X$. Eq.(4) generalizes eq.(2) to states described by density matrices.

III. INCLUSION OF RESIDUAL ERRORS AND THE QCC

We now extend Kitaev's model by allowing for the inevitable survival of residual errors in any realistic implementation of a quantum computation, even upon successful application of fault-tolerance techniques. In other words, we will extend the analysis to include fault-tolerant operation, but in such a way as to ensure a prescribed success probability to achieve the correct final result of the overall computation.

We consider a function F , and an implementation of an overall computation (including a quantum computation followed by a final measurement) that is intended to calculate F for some value of the input. The relationship between the *ideal*, error-free quantum computation, $G(\rho) = U\rho U^\dagger$, defined on the logical Hilbert space, H_{logical} , and the *actual* dynamical map, P , implemented

by the physical quantum computer on the computational Hilbert space, H_{comp} , can be described by the following diagram:

$$\begin{array}{ccc} \mathbf{T}(H_{\text{comp}}) & \xrightarrow{P} & \mathbf{T}(H_{\text{comp}}) \\ \mathcal{M}_{\{1 \rightarrow c\}} \uparrow & \alpha & \downarrow \mathcal{M}_{\{c \rightarrow 1\}} \\ \mathbf{T}(H_{\text{logical}}) & \xrightarrow{G} & \mathbf{T}(H_{\text{logical}}) \end{array} \quad (5)$$

where the superoperators $\mathcal{M}_{\{1 \rightarrow c\}}$ and $\mathcal{M}_{\{c \rightarrow 1\}}$ are *linking maps* that are mathematically required to relate states in the logical space H_{logical} to states in the computational space H_{comp} . This is because, in general, $H_{\text{logical}} \neq H_{\text{comp}}$ [7].

We formally express the content of Diagram (5) *via* a relation between an idealized (G) and actual (P) quantum computation:

$$\|\mathcal{M}_{\{c \rightarrow 1\}}(P \cdot (\mathcal{M}_{\{1 \rightarrow c\}}(\rho))) - G(\rho)\|_1 \leq \alpha, \quad (6)$$

where $\|\cdot\|_1$ signifies the Schatten 1-norm. The quantity on the left-hand-side of this inequality furnishes a measure of the inaccuracy of the actual implementation of G by P . It tells us how well a practical quantum computing device actually implements an ideally defined quantum computation. We will refer to the left-hand-side of this inequality as the *implementation inaccuracy* of the quantum computation. The quantity, α , on the right-hand-side of this inequality is a bound on the implementation inaccuracy. The value of α is *prescribed* as a requirement on the performance of the quantum computation. Thus, the diagram states that P can implement the idealized, perfect quantum computation, G , with an inaccuracy no greater than α . We refer to the entire inequality as the *Quantum Computer Condition* (QCC).

Thus, we have that Diagram (3) connects the ideal quantum computation, G , to the calculation of F , and, separately, we have that Diagram (5) connects the ideal quantum computation, G , to the actual implementation of the quantum computation, P , that is realized by a practical physical device. To establish the utility of practical quantum computers, what is needed now is a relation that connects the actual, device-implemented quantum computation, P , to the intended calculation of F . By analogy with eq.(4), we seek an expression of the form

$$\text{tr}(\text{O}_{\text{dm}}(P^{\mathcal{M}}(\text{I}_{\text{dm}}(x)))) \equiv \text{tr}\left(\sqrt{E_{F(x)}}P^{\mathcal{M}}(\text{I}_{\text{dm}}(x))\sqrt{E_{F(x)}}\right) > 1 - p', \quad (7)$$

for some p' , where $P^{\mathcal{M}} \equiv \mathcal{M}_{\{c \rightarrow 1\}}P\mathcal{M}_{\{1 \rightarrow c\}}$. This is established with the following calculation:

$$\begin{aligned} \text{tr}(\text{O}_{\text{dm}}(P^{\mathcal{M}}(\text{I}_{\text{dm}}(x)))) &= \text{tr}\left(\sqrt{E_{F(x)}}P^{\mathcal{M}}(\text{I}_{\text{dm}}(x))\sqrt{E_{F(x)}}\right) \\ &= \text{tr}\left(\sqrt{E_{F(x)}}\{P^{\mathcal{M}}(\text{I}_{\text{dm}}(x)) - U\text{I}_{\text{dm}}(x)U^\dagger\}\sqrt{E_{F(x)}}\right) \end{aligned}$$

$$\begin{aligned}
& + \operatorname{tr} \left(\sqrt{E_{F(x)}} U I_{\mathbf{dm}}(x) U^\dagger \sqrt{E_{F(x)}} \right) \\
& > -\alpha + 1 - p,
\end{aligned} \tag{8}$$

where we have used the property of the Schatten 1-norm,

$$\left| \operatorname{tr} \left(\sqrt{E_{F(x)}} \{ P^{\mathcal{M}}(I_{\mathbf{dm}}(x)) - U I_{\mathbf{dm}}(x) U^\dagger \} \sqrt{E_{F(x)}} \right) \right| \leq \| P^{\mathcal{M}}(I_{\mathbf{dm}}(x)) - U I_{\mathbf{dm}}(x) U^\dagger \|_1, \tag{9}$$

and the condition

$$\| P^{\mathcal{M}}(I_{\mathbf{dm}}(x)) - U I_{\mathbf{dm}}(x) U^\dagger \|_1 = \| \mathcal{M}_{\{c \rightarrow l\}}(P \cdot (\mathcal{M}_{\{l \rightarrow c\}}(I_{\mathbf{dm}}(x)))) - G(I_{\mathbf{dm}}(x)) \|_1 \leq \alpha. \tag{10}$$

The physical meaning of this result can be transparently expressed by combining the content of Diagrams (3) and (5) to yield

$$\begin{array}{ccc}
\mathbf{T}(H_{\text{comp}}) & \xrightarrow{P} & \mathbf{T}(H_{\text{comp}}) \\
\tilde{I}_{\mathbf{dm}} \uparrow & \alpha + p & \downarrow \tilde{O}_{\mathbf{dm}} \\
X & \xrightarrow{F} & Y
\end{array} \tag{11}$$

This diagram is nearly commutative in the sense that

$$\begin{aligned}
\operatorname{tr} (O_{\mathbf{dm}} (P^{\mathcal{M}}(I_{\mathbf{dm}}(x)))) & \equiv \operatorname{tr} (\tilde{O}_{\mathbf{dm}}(P(\tilde{I}_{\mathbf{dm}}(x)))) \\
& > 1 - (p + \alpha),
\end{aligned} \tag{12}$$

where $\tilde{I}_{\mathbf{dm}} \equiv \mathcal{M}_{\{l \rightarrow c\}} \circ I_{\mathbf{dm}}$, $\tilde{O}_{\mathbf{dm}} \equiv O_{\mathbf{dm}} \circ \mathcal{M}_{\{c \rightarrow l\}}$. We have thus established eq.(7) with $p' \equiv p + \alpha$. Thus the total probability of failure that the overall computation yields the correct *final* result is bounded by the sum of two terms: (1) an upper bound on the failure probability for the overall computation when the quantum computation is perfectly implemented without errors, and (2) an upper bound on the implementation inaccuracy of the actual quantum computation implemented by a practical device. This result is both intuitively simple and technically subtle. It is not surprising that the total failure probability for the overall computation to yield the correct final result is bounded by quantities that describe the two sources of error. However, it is not immediately obvious that these quantities should combine in such a simple way. On the one hand, the quantities p and p' represent bounds on probabilities for obtaining certain outcomes from a measurement of the final quantum state. On the other hand, the quantity α is a bound, not on a probability, but on the normed difference between the states resulting from idealized and actual quantum computations, respectively.

Diagram (11) and eq.(12) relate the actual implementation P of the quantum computation to the intended calculation of F . They state the criterion for a realization, P , of a quantum computer, operating fault-tolerantly in the presence of residual errors, to correctly implement an instance of the overall computation. If p' is sufficiently small a majority vote algorithm will successfully identify

the correct outcome $F(x)$. For example, in the case Y is binary, the calculation of F succeeds by majority voting if $p' < 1/2$.

Note that in the idealized limit in which error correction *perfectly* and *permanently* removes all residual errors (*i.e.*, in the limit $\alpha = 0$), our result reduces to the corresponding result of the Kitaev model (*i.e.*, eq.(12) reduces to eq.(4)).

IV. APPLICATIONS TO PRACTICAL SPECIFICATIONS FOR FAULT TOLERANCE

A. General Result Relating Fault Tolerance Theory to the Overall Computation

The above analysis, in which the constraints of fault tolerance are explicitly combined with those of the Kitaev model, has important practical applications to the specification of error tolerances for quantum computers. The designer of a quantum computer wishes to achieve some upper bound p' on the probability that the final measurement of the final quantum state produced by the quantum computation *fails* to yield the correct final result for the overall computation. As discussed above, there is some bound, p , on the inherent probability that the overall computation will fail even if the quantum computation is perfectly implemented. As discussed above, this is due to the intrinsic non-determinism inherent in the final measurement of the final quantum state resulting from the execution of the quantum computation, and it is an abstract property of the algorithm itself. Our result shows that the implementation of the algorithm by a realistic, *i.e.* noisy, quantum computer will meet the designer's success criterion provided that the implementation inaccuracy satisfies $\| \mathcal{M}_{\{c \rightarrow l\}}(P \cdot (\mathcal{M}_{\{l \rightarrow c\}}(\rho))) - G(\rho) \|_1 \leq p' - p$. Since the ideal failure probability bound, p , is a characteristic of the ideal quantum algorithm itself, this result effectively apportions the allowable noise in a quantum computation between a component, p , due to the inherent quantum mechanical indeterminacy associated to the measurement of the final quantum state, and a component, α , associated to the dynamics of the quantum computation itself,

which includes other sources of noise, such as decoherence. Eq.(6) thus provides a success criterion for the design and implementation of a fault-tolerant quantum computation that, upon measurement of the final quantum state of which, produces the *correct final result* for the overall computation.

As an example of how our result can be applied in order to achieve practical constraints on design parameters, suppose we wish to build a fault tolerant quantum computer which provides the correct solution to some specific problem (*e.g.* factoring a large number) with probability $1 - \hat{p}$ or better. That is, \hat{p} is the required upper bound on the probability that the overall computation produces the wrong final result. From eq.(12) we infer that the failure probability of the overall computation *as implemented* is bounded by $p + \alpha$. We therefore require

$$p + \alpha = \hat{p} . \quad (13)$$

The quantity p is an intrinsic characteristic of the quantum algorithm, and can always, in principle, be determined. We therefore require

$$\alpha = \hat{p} - p , \quad (14)$$

which is a bound on the implementation inaccuracy of the quantum computation, sufficient to meet the prescribed success requirement for the overall computation. From eq.(6), we see that this criterion is met if

$$\|\mathcal{M}_{\{c \rightarrow 1\}}(P \cdot (\mathcal{M}_{\{1 \rightarrow c\}}(\rho))) - G(\rho)\|_1 \leq \hat{p} - p \quad (15)$$

for all ρ .

Techniques of fault tolerance theory can be used to determine the probability ϵ_{QC} that the quantum computation fails to produce the desired final quantum state. We therefore write the final state of the *logical* qubits resulting from the actual, practically-implemented, quantum computation as

$$\mathcal{M}_{\{c \rightarrow 1\}}(P \cdot (\mathcal{M}_{\{1 \rightarrow c\}}(\rho))) = (1 - \epsilon_{QC})G(\rho) + \epsilon_{QC}\rho_{err} , \quad (16)$$

where $G(\rho)$ would be the result of an idealized quantum computation, ρ_{err} arises from errors, and the *entire right hand side* of eq. (16) is the state that results when errors occur. Eq. (15) then becomes

$$\|\epsilon_{QC}[\rho_{err} - G(\rho)]\|_1 = \epsilon_{QC}\|\rho_{err} - G(\rho)\|_1 \leq \hat{p} - p . \quad (17)$$

Since,

$$\|\rho_{err} - G(\rho)\|_1 \leq 2 , \quad (18)$$

eqs. (17), (15), and therefore (13) will be satisfied provided

$$\epsilon_{QC} \leq \frac{1}{2}(\hat{p} - p) . \quad (19)$$

This is a new, quite general result relating the fault tolerance theoretic parameter ϵ_{QC} to the constraints we have

derived, which ensure that the overall computation yield the correct final result with some prescribed success probability. This general inequality at once combines the constraints dictated by the QCC, which apply to the dynamics of the quantum computation, with those coming from the Kitaev model, which apply to the measurement, in terms of the numerical parameters of standard fault tolerance theory, which apply only to gate failure probabilities.

B. Direct Example: Calculation of the Concatenation Level Function

As an example of how our general result, Eq.(19), can be applied to the specification of practical fault-tolerance design parameters, we consider the fault tolerance approach described in [4], in which a concatenated quantum error correcting code is applied so as to reduce the failure probability at each level of concatenation. The failure probability for a single logical gate scales roughly as [4]

$$\epsilon_N \simeq \epsilon_{th} \left(\frac{\epsilon_0}{\epsilon_{th}} \right)^{2^N} , \quad (20)$$

where ϵ_0 is the probability of failure for elementary gates, ϵ_{th} is the fault tolerance threshold, and ϵ_N is the failure probability of the gate at the N th level of concatenation. For simplicity, in this example we assume that this relation is exact, that it applies equally to all gates, and that no other sources of error are present. If the quantum computation is comprised of \mathcal{N} logical gates, then the failure probability for the quantum computation to yield the desired final quantum state is

$$\epsilon_{QC} \simeq \mathcal{N}\epsilon_{th} \left(\frac{\epsilon_0}{\epsilon_{th}} \right)^{2^N} . \quad (21)$$

Thus, in order to ensure that we obtain the correct final result to the overall computation with sufficient probability of success, we make use of our general result in eq.(19) to require that

$$\epsilon_{QC} \simeq \mathcal{N}\epsilon_{th} \left(\frac{\epsilon_0}{\epsilon_{th}} \right)^{2^N} \leq \frac{1}{2}(\hat{p} - p) \quad (22)$$

is satisfied. From this we infer that the number of levels of concatenation sufficient to guarantee that the overall computation performs as required is given by:

$$N \gtrsim \log_2 \frac{\ln \frac{2\mathcal{N}\epsilon_{th}}{\hat{p}-p}}{\ln \frac{\epsilon_{th}}{\epsilon_0}} . \quad (23)$$

This is a sufficient, not a necessary, condition. Alternatively, for a given level of concatenation, we could just as well derive a requirement on the error probability ϵ_0 for elementary gates. In other words, this result enables us

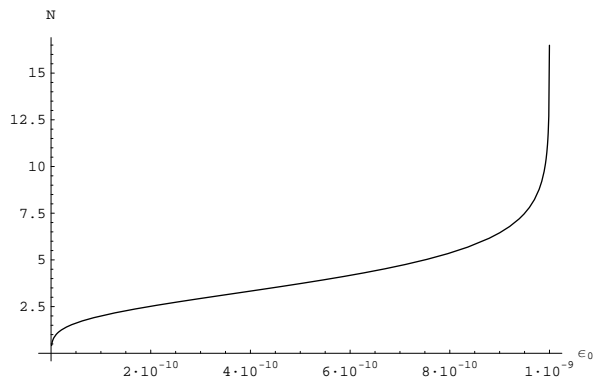


FIG. 2: Levels of concatenation, N , required to meet the performance criterion as a function of elementary gate error probability, ϵ_0 . The performance criterion is to achieve the correct final result for the overall computation (not simply the desired final quantum state) with some prescribed success probability. In this example the success probability (\hat{p}) is prescribed to be 0.6. The bound on error probability associated to the measurement following ideal quantum computation (p) is taken to be 0.2. The number of gates (N) is 10^{12} , and the error threshold (ϵ_{th}) is taken to be 10^{-9} .

to explore the tradeoff between adding additional levels of concatenation and improving the performance of elementary gates in order to achieve a given performance criterion in terms of the reliability of the final result. An example of such a tradeoff curve is shown in Figure 2. We stress that the points on the curve in Figure 2 do not merely represent the amount of concatenation needed in order that the *quantum computation* produce a particular final quantum state, but rather, fully incorporate the conditions that will ensure a successful *overall computation*.

We emphasize that the role of standard fault tolerance theory in our calculation is solely to express the error probability for the quantum computation in terms of elementary gate errors. Standard fault tolerance theory does not provide estimates of (nor bounds on) the implementation inaccuracy as defined in eq.(6), and, consequently, would not have allowed us to calculate the number of levels of concatenation required in order to ensure that the overall computation produces the correct final result with the prescribed probability of success.

V. CONCLUSION

In this paper we have introduced a methodology for determining design parameters for a quantum computer, the values of which ensure that an overall computation of interest, comprised of an initial purely quantum computation followed by a measurement of the final quantum state produced by the quantum computation, yields a correct *final result* with some prescribed probability of success, as opposed to merely ensuring that the de-

sired *final quantum state* is obtained. Thus, our method enables the determination of *complete* quantum computational resource requirements associated to the actual solution of practical problems

Our method fully accounts for two sources of error that may affect the final result of the overall computation: (1) errors which arise in the course of the quantum computation itself, and (2) errors due to the indeterminacy intrinsic to the final measurement which follows the quantum computation. Standard fault tolerance theory addresses the first type of error, but does not consider the second. Kitaev's model addresses the second type of error, but not the first. We have extended both standard fault tolerance theory and Kitaev's model, and have combined them, in order to yield quantitative results that reveal the interplay between the two. Although the analysis in this paper has been presented in the framework of the circuit paradigm for quantum computing, it is straightforward to apply our results to other paradigms, including the cluster state and adiabatic paradigms [8].

As a specific example of the practical application of our approach, we have explicitly calculated the number of levels of error correction concatenation needed to achieve a correct final result for the overall computation with some prescribed success probability. Extensions of the current calculation will include associating different failure probabilities to different gates, as well as considering additional refinements dictated by imposing the QCC on the overall dynamics of the quantum computer [8].

Acknowledgments

The authors thank F. Javier Thayer for his contributions to the early phases of this research. They also thank Stephen P. Pappas and Anthony Donadio for their input. This research was supported under MITRE Technology Program Grant 07MSR205.

-
- [1] P.W. Shor, SIAM Journal on Computing **26**, 1484 (1997).
 - [2] A. Kitaev, Russian Mathematical Surveys, **52**, 1191 (1997).
 - [3] For simplicity we do not consider errors in the preparation of the initial quantum state in the present paper.
 - [4] J. Preskill, Proc. Roy. Soc. Lond. A **454**, 385 (1998).
 - [5] D. A. Lidar, D. Bacon, K. B. Whaley, Phys. Rev. Lett. **82**, 4556 (1999).
 - [6] M. Nielsen, I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge (2000).
 - [7] This can be illustrated with the example of the 7-qubit Steane code, for which H_{comp} is comprised of 7 computational qubits for every logical qubit in H_{logical} . We note that the linking maps $\mathcal{M}_{\{1 \rightarrow c\}}$ and $\mathcal{M}_{\{c \rightarrow 1\}}$ do not represent transformations carried out by physical devices. The physical encoding and decoding operations implied by the use of, *e.g.*, the Steane code, must be performed by physical devices and are therefore subject to noise. Such physical encoding and decoding operations are properly included in P and not in the linking maps. In the context of the Steane code, one may think of the linking maps as representing the formal extension of Hilbert space to include the necessary ancilla qubits.
 - [8] G. Gilbert, M. Hamrick, and Y.S. Weinstein, work in progress.